HAL open science

Occupancy grids from stereo and optical flow data

Christophe Braillon, Cédric Pradalier, Kane Usher, Jim Crowley, Christian Laugier

To cite this version:

Christophe Braillon, Cédric Pradalier, Kane Usher, Jim Crowley, Christian Laugier. Occupancy grids from stereo and optical flow data. Proc. of the Int. Symp. on Experimental Robotics, Jul 2006, Rio de Janeiro (BR), France. inria-00182020

HAL Id: inria-00182020

https://inria.hal.science/inria-00182020v1

Submitted on 24 Oct 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Occupancy grids from stereo and optical flow data

C. Braillon[1], C. Pradalier[2], K. Usher[2], J.L. Crowley[1], and C. Laugier[1]

[1] INRIA, Grenoble, France
firstname.lastname@inrialpes.fr
[2] CSIRO ICT Centre, Brisbane, Australia
firstname.lastname@csiro.au

Summary. In this paper, we propose a real-time method to detect obstacles using theoretical models of the ground plane, first in a 3D point cloud given by a stereo camera, and then in an optical flow field given by one of the stereo pair's camera.

The idea of our method is to combine two partial occupancy grids from both sensor modalities with an occupancy grid framework. The two methods do not have the same range, precision and resolution. For example, the stereo method is precise for close objects but cannot see further than 7 m (with our lenses), while the optical flow method can see considerably further but has lower accuracy.

Experiments that have been carried on the CyCab mobile robot and on a tractor demonstrate that we can combine the advantages of both algorithms to build local occupancy grids from incomplete data (optical flow from a monocular camera cannot give depth information without time integration).

1 Introduction

This work takes place in the general context of mobile robots navigating in open and dynamic environments. Computer vision for ITS (Intelligent Transport Systems) is an active research area [7]. One of the key issues of ITS is the ability to avoid obstacles. This requires a method to perceive them.

In this article we address the problem of obstacle sensing through their motion (optical flow) in an image sequence. The perceived motion can be caused either by the obstacle itself or by the motion of the camera (which is the motion of the robot in the case of a camera fixed on it). We also use a stereo camera to improve the range (for short sight) of the resulting sensor.

Many methods have been developed to find moving objects in an image sequence. Most of them use a fixed camera and use background subtraction (for example in [14] and [6]).

Recently, in [16], a new approach to obstacle avoidance has been developed, based on ground detection by finding planes in images. The weak point of this method is that the robot must be in a static environment.

Model based approaches using ego-motion have been demonstrated in [10], [15]. The first one detects the ground plane by virtually rotating the camera and visually estimating the ego-motion. The second one uses dense stereo and optical flow to find moving objects and robot ego-motion. These two methods have a large computational cost as several successive calculations (stereo, optical flow, ego-motion, ...) are required.

In this paper, we demonstrate that we can model the motion of the ground plane and determine the location of the obstacles, using the motion of the camera [2]. Moreover we show that the stereo data improves the accuracy for short range obstacle detection.

One key point in this method is that we do not compute explicitly the optical flow of the image at any time. Optical flow computation is very expensive in terms of CPU time, is inaccurate and sensitive to noise. In general we can see in the survey led by Barron et al. ([1]) that the accuracy of optical flow computation is linked to the computational cost. We model the expected optical flow (which is easy and quick to compute) to get rid of the inherent noise and the time consuming optical flow step. As a consequence we are able to demonstrate robust and real-time obstacle detection.

2 Model-based obstacle detection (stereo and optical flow)

The two algorithms we will use in the next two parts are model-based approaches. In a first step, we model our expected observation in the sensor space (in this article we focus on the ground plane model). In the second step we cluster observation into two sets: those who matches the model – points on the ground plane – and those who don't: the obstacles.

Let $n \in \mathbb{N}$ be the dimension of the observation space and $m \in \mathbb{N}$ a number of parameters. The model $\mathcal{F}$ with m parameters is a function $(\mathbb{R}^n \times \mathbb{R}^m) \rightarrow \mathbb{R}^p$ and is defined by the relation:

$$\mathcal{F}(Z, P) = 0, P \in \mathbb{R}^m \text{ and } Z \in \mathbb{R}^n$$

Given a model $\mathcal{F}$ of our observations we try to extract the parameter set $P \in \mathbb{R}^m$ from a set of observed data $(Z_i)_{i=1\cdots k}$ by minimising the error in the system:

$$\mathcal{F}(Z_i, P) = 0, \forall i \in \{1 \cdots k\}$$

This minimisation can be done by performing a Least Mean Squares (LMS) minimisation. However, for better outlier rejection, here we use a Least Median Squares technique with the minimisation performed by the Nelder-Mead Simplex search [13].

Once the parameters are retrieved, we can cluster the observation in two sets (observations that match the model and observations that do not).

In the next part, we will use a ground plane model in both optical flow and 3D world spaces. In that case, the parameter sets will give us the position of the camera (with respect to the ground plane) and its ego-motion. The observation will be clustered in two sets: the ground plane and the obstacles.

2.1 Stereo obstacle detection

For obstacle detection using the stereo camera, we take the point cloud data generated from the stereo images, find the dominant plane in the point cloud (which should be the ground-plane) using the Least Median Squares method, finally the point cloud is converted into an occupancy grid using the following principles.

The observation space using the point cloud data is $\mathbb{R}^3$. Working in Cartesian space, the ground plane can be described by a set of four parameters ($\mathbb{R}^4$). The model is then expressed as:

$$\mathcal{F}(Z, P) = p_1 x + p_2 y + p_3 z + p_4$$

where $Z = (x, y, z)$ and $P = (p_1, p_2, p_3, p_4)$. Figure 1 shows an example of the stereo point cloud data together with the plane fitted to the data.

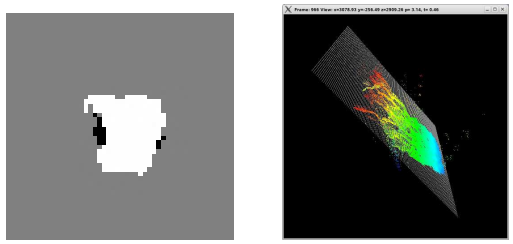
Having found the ground plane, we can now estimate the camera height, roll and tilt, and compare these to the 'expected' values (from knowledge of the camera mounting position). We can also populate the occupancy grid using the idea that points not on the ground plane (at least within a tolerance) must belong to an obstacle. That is, we can calculate the distance of each point from the ground plane using the following equation:

$$e_i = p_1 x_i + p_2 y_i + p_3 z_i + p_4$$

where the subscript i denotes the ith point in the cloud. If this distance exceeds a threshold, then the point contributes to the evidence that the ground plane cell to which the point belongs to contains an obstacle. Otherwise, if the distance is below the threshold, the point contributes to the evidence that the cell is free. Figure 1 illustrates an example occupancy grid generated using this method. In this figure, an 'empty' cell is represented by black, an 'occupied' cell is represented by white, and 'unknown' cells are represented by grey. The viewpoint is from the bottom of the image. Note the difference to, for example, a scanning laser generated occupancy grid, which physically can't 'see' behind objects.

2.2 Optical flow obstacle detection

By definition, an optical flow field is a vector field that describes the velocity of pixels in an image sequence. The first step of our method is the modelling of the optical flow field for our camera. The camera model we used is the classical pinhole camera model.



(a) (b)

Fig. 1. (a) Example of an occupancy grid generated from the stereo data and plane fitting process. The viewpoint is from the bottom centre of the image. (b) Example of the stereo point cloud data. Also shown is the plane fitted to the data.

Naive first approach

The parametrisation of our model can be found in [12], it says that only 8 parameters (parameter space is $\mathbb{R}^8$ and $P = (p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8)$) are needed to fully describe the visual motion of a plane. We call $Z = (u, v, f_u, f_v)$ an observation of an optical flow vector $\mathbf{f} = (f_u, f_v)$ at pixel (u, v). Therefore the observation space is $\mathbb{R}^4$. We can write the new model as follows:

$$\mathcal{G}(Z, P) = \begin{pmatrix} (p_1 - f_u) + p_2 u + p_3 v + p_4 u^2 + p_5 uv \\ (p_6 - f_v) + p_7 u + p_8 v + p_4 uv + p_5 v^2 \end{pmatrix}$$

Using this method for optical flow requires a good accuracy on the ground plane to evaluate the parameters of the ground plane. Indeed, the part of the flow field we want to model is the ground plane. Therefore we need a good accuracy on its optical flow. Moreover we want to respect our real-time constraint. Thus we need a method that perform accurate optical flow computation in real-time.

We reviewed all the characteristics of various optical flow method described in [1] but no method was really appropriate (either inaccurate or slow). We used brand new optical flow computation methods developed by Bruhn, Weickert *et al.* (in [3], [4], [5], ...). They are the most accurate real-time method we found. They give good information on uniform surfaces (they use a global constraint like in Horn and Schunck technique [9] to compute the flow field on uniform surfaces).

Even with state-of-the-art techniques, the optical flow we compute on the ground plane is inaccurate. Indeed, the ground plane is often poorly-textured (asphalt on the road) and is a large part of the image. We can see on figure 2 that the optical flow of the ground plane is inaccurate.

(a) (b)

Fig. 2. Figure (a) is an image from a video sequence where the camera is translating and the pedestrian is moving in front of the robot. Figure (b) is the corresponding optical flow computed with [4]. Note the incorrect optical flow vectors on the ground plane.

Odometry based optical flow model

Having observed that the optical flow computation does not give results good enough for our optimisation, we proposed a reverse method which tries to match an optical flow model given by the odometry data to the image.

To describe our model, we will use the projective geometry formalism. We will call (u, v, w) the homogeneous coordinates of a pixel in the image, $\mathcal{H}$ the homography matrix that projects one point on the ground plane in the image and $\dot{\mathcal{H}}$ its derivative.

The projection equation is (1) from which we can infer (2) by differentiating. Finally, we can obtain the optical flow vector $\mathbf{f}$ for the pixel at Euclidean coordinates (u, v) (equation (3)).

$$(\dot{u}, \dot{v}, \dot{w})^T_{Image} = \dot{\mathcal{H}} \, (X, Y, 1)^T_{Image} \tag{1}$$

$$(\dot{u}, \dot{v}, \dot{w})^T_{Image} = \dot{\mathcal{H}} \mathcal{H}^{-1} \, (u, v, 1)^T_{Image} \tag{2}$$

$$\mathbf{f}(u, v) = (\dot{u} - u\dot{w}, \dot{v} - v\dot{w})^T \tag{3}$$

Finally from equations (2) and (3) we can express the theoretical optical flow vector for each pixel in the image (with the assumption that each pixel is in the ground plane).

The homography matrix, $\mathcal{H} = (h_{i,j})_{i,j=1..3}$ and its derivative $\dot{\mathcal{H}}$ are evaluated using the position of the camera (c_x, c_y, c_z), its orientation ϕ, the odometry-given motion of the camera v and ω (linear and angular velocity).

$$\begin{aligned}
h_{1,1} &= u_0 \cos\phi & \dot{h}_{1,1} &= \alpha_u \omega \\
h_{1,2} &= -\alpha_u & \dot{h}_{1,2} &= u_0 \omega \cos\phi \\
h_{1,3} &= \alpha_u + u_0 \left(-\cos\phi + c_z \sin\phi\right) & \dot{h}_{1,3} &= -u_0 v \cos\phi \\
h_{2,1} &= -\alpha_u \sin\phi + v_0 \cos\phi & \dot{h}_{2,1} &= 0 \\
h_{2,2} &= 0 & \dot{h}_{2,2} &= \left(-\alpha_v \sin\phi + v_0 \cos\phi\right) \omega \\
h_{3,1} &= \cos\phi & \dot{h}_{3,1} &= 0
\end{aligned}$$

$$h_{2,3} = \alpha_v (\sin\phi + c_z \cos\phi) + v_0 (-\cos\phi + c_z \sin\phi) \qquad \dot{h}_{2,3} = (\alpha_v \sin\phi - v_0 \cos\phi)\, v$$

$$h_{3,2} = 0 \qquad \dot{h}_{3,2} = \omega \cos\phi$$

$$h_{3,3} = -\cos\phi + c_z \sin\phi \qquad \dot{h}_{3,3} = -v \cos\phi$$

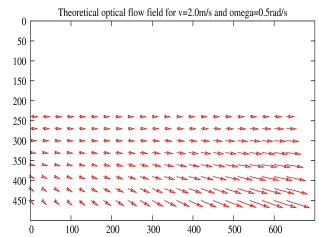


Fig. 3. Example of theoretical optical flow field for a moving robot with a velocity of 2 m.s^{-1} and a rotation speed of 0.5 rad.s^{-1}

Figure 3 shows the result of our model for a camera at position $c_x = 1.74$ m, $c_y = 0$ m, $c_z = 0.83$ m and $\phi = 0$ rad. The model is valid only below the horizon line whose equation is: $y = v_0 - \alpha_v \tan\phi$. Therefore there is no flow vector above the horizon line.

Once theoretical optical flow field is computed, we can match it to the observed data. We use two consecutive images and try to match one pixel in the previous image to the corresponding theoretical pixel in the current image. The matching is done by computing an SSD (Sum of Squared Differences) measure. An example of the model fitting based on SSD correlation can be seen on figure 4

(a) (b) (c)

Fig. 4. Result of the optical flow clustering. Image (a) and (b) are two consecutive frame of a video sequence. Subfigure (c) is the result of the SSD matching. The dashed area is the one where the model does not apply

3 Vision-oriented data fusion

In this section, we will now concentrate on fusing together information coming from optical flow and stereo vision. Among data fusion methods, occupancy grids are used often to deal with dense data set. The occupancy grid formalism has been developped originally for 2D environment but can readily be adapted to 3D data such as those coming from a camera. Unfortunately the computational complexity of 3D grids is prohibitive for real-time application. In this section we will describe how 3D data from optical flow and stereo-vision are fused using 2D-only occupancy grid.

3.1 3D camera sensor model

It is difficult to deal with cameras in an occupancy grid framework. Indeed, one pixel of the image can correspond to an infinite set of 3D world points. This set of points is known as projective line.

The set of projective lines corresponding to all the image pixels is a pyramid (dimension 3). Therefore we need to express the occupancy grids in a 3D space.

Figure 5 shows the camera model we use. The shape on the image can correspond to the dashed pyramid. Saying that a pixel in the image belongs to an obstacle, means that the projective line is potentially occupied.

3.2 2D projected model

The occupancy grid framework is very expensive when it comes to a 3D space. The idea is to project the 3D occupancy grid on the ground plane. This projection respects the semantics of occupancy grids and translates the incompleteness of the monocular camera model into uncertainty.

The projection of a 3D occupancy grid $C^{3D}_{i,j,k}$ into a 2D occupancy grid $C_{i,j}$ is defined in our work as:

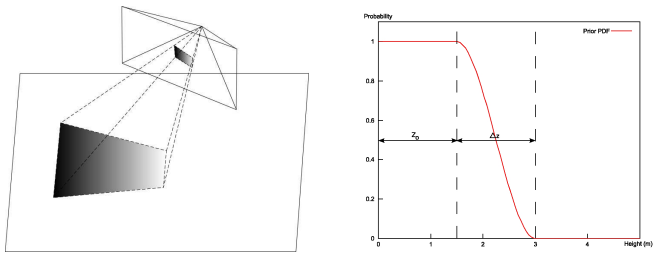
$$C_{i,j} = \max_k \left(p_k \times C^{3D}_{i,j,k} + (1 - p_k)\,\frac{1}{2} \right) \tag{4}$$

In equation (4) we use the maximum operator to have an occupancy grid as safe as possible. Indeed, we impose that the probability for a cell to be occupied is the maximum of the probabilities for all the 3D cells on top of the 2D ground cell, which means that we do not risk saying that a 2D cell is occupied if a 3D cell over it is occupied.

The term p_k is *a priori* knowledge. It gives *a priori* knowledge on the vertical distribution of the obstacles. The value 1 means a strong confidence and the value 0 means that no obstacle can be at the given height.

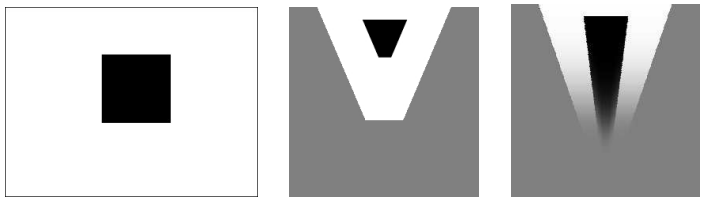
We imposed the following function for p_k (see figure 5):

$$p_k = \begin{cases} 1 & \text{if } k \leq z_0 \\ 2\left(\frac{z-z_0}{\Delta z}\right)^3 - 3\left(\frac{z-z_0}{\Delta z}\right)^2 + 1 & \text{if } k \in]z_0, z_0 + \Delta z] \\ 0 & \text{elsewhere} \end{cases} \tag{5}$$



(a) (b)

Fig. 5. (a) Camera model, the shape in the image is projected on the ground plane. The dashed pyramid corresponds to the potentially occupied space. (b) Graph of the function p_k defined in equation (5). We can see on that figure that the highest probability is given to low height.



(a) (b) (c)

Fig. 6. (a) is a square on the image, (b) is its basic projection on the ground plane, (c) is the 3D model projected on the ground plane.

Figure 6 shows the different steps of the pyramid projection. On subfigure (c) we can see that the shape is fading out, this is because the sensor model we used gives more probability to the obstacles close to the ground.

To fuse all the sensor modalities we use an occupancy grid framework [8], [11], to express the fusion in formal probabilistic terms.

4 Experimental result

On figure 7 (b), we can clearly see the pedestrian moving in front of the camera. We can also see the blue car in the back. The relative importance

of the car and the pedestrian is due to the distance between the camera and them. The closer to the camera the objects are, the bigger their optical flow is. In a future work we could improve this point by exploring the possibility of normalizing the SSD by the optical flow. This would result in a better ratio between far and close obstacles.

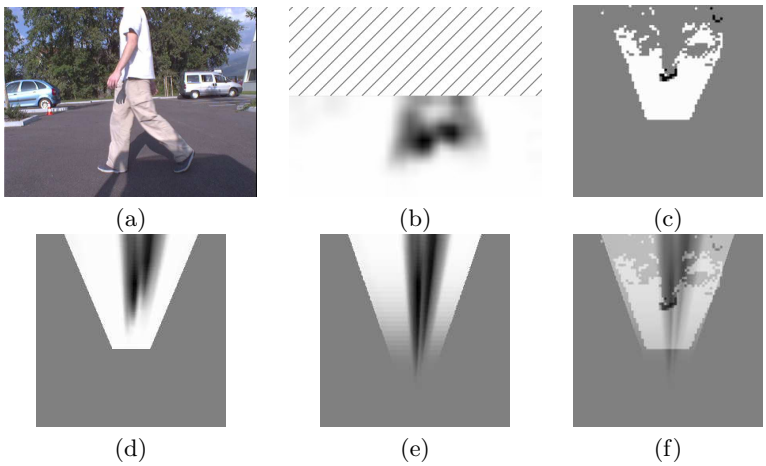


Fig. 7. (a) is the left image of the stereo pair, (b) is the detected obstacle from optical flow, (c) is the occupancy grid generated from the point cloud, (d) is the projection of (b) on the ground plane, (e) is the improved model presented in 3.2 and (f) is the fusion of the two occupancy grids (c) and (e)

Subfigure 7 (c) shows the pedestrian in the middle of the grid. The grid is not very dense after 3 m but give information until 7 m.

Finally, subfigure 7 (f) shows the global result of the fusion. We used the same confidence for both algorithms. We can see that the area where the stereo does not provide a dense information are supplemented by the optical flow algorithm. The area where the pedestrian is, is also reinforced. The false detection on the top right of the grid is minimised. After the fusion step, this false detection has a probability which means the occupancy is unknown. The cells in front of the obstacle are a little degraded.

5 Conclusion and future work

In this paper, we proposed a real-time method to detect obstacles using theoretical models of the ground plane using the 3D point cloud given by a stereo camera, and an optical flow field given by one of the stereo pair's camera.

The performance of the global process is better than the stereo detection or the optical flow detection alone. We could improve the quality of the occupancy grid by adding more sensors (other cameras, laser range finders, ...) and/or more camera modalities (colour segmentation, ...).

The next step will be to perform time integration to remove some ambiguities (especially ambiguities related to monocular camera algorithms).

6 Acknowledgements

This work was done in the context of a cooperation between the CSIRO ICT Centre, Autonomous System Lab, in Brisbane (Australia) and the INRIA Rhône-Alpes in Grenoble (France).

References

1. J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Performance of optical flow techniques. In *IJCV*, volume 12, pages 43–77, 1994.
2. C. Braillon, C. Pradalier, J.L. Crowley, and C. Laugier. Real-time moving obstacle detection using optical flow models. 2006.
3. T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. 3024:25–36, may 2004.
4. A. Bruhn, J. Weickert, C. Feddern, T. Kohlberger, and C. Schnorr. Variational optical flow computation in real-time. 14/5:608–615, 2005.
5. A. Bruhn, J. Weickert, and C. Schnrr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. 61/3:211–231, 2005.
6. R. Collins, A. Lipton, H. Fujiyoshi, and T. Kanade. Algorithms for cooperative multisensor surveillance. *Proc. of the IEEE*, pages 1456–1477, October 2001.
7. E.D. Dicksmanns. The development of machine vision for road vehicles in the last decade. 2002.
8. A. Elfes. Using occupancy grids for mobile robot perception and navigation. 22(6):46–57, 1989.
9. B.K.P. Horn and B.G. Schunck. Determining optical flow. In *artificial intelligence*, volume 17, pages 185–203, 1981.
10. Q. Ke and T. Kanade. Transforming camera geometry to a virtual downward-looking camera: robust ego-motion estimation and ground layer detection. 2003.
11. K. Konolige. Improved occupancy grids for map building. 4:351–367, 1997.
12. H.C. Longuet-Higgins. The visual ambiguity of a moving plane. In *Royal Society London*, London, Great Britain, 1984.
13. J. A. Nelder and R. Mead. A simplex method for function minimization. pages 308–313, 1965.
14. C. Stauffer and W.E.L. Grimson. Adaptative background mixture models for real-time tracking. January 1998.
15. A. Talukder and L. Matthies. Real-time detection of moving objects from moving vehicle using dense stereo and optical flow. October 2004.
16. K. Young-Geun and K. Hakil. Layered ground floor detection fo vision-based mobile robot navigation. In *ICRA*, pages 13–18, New Orleans, april 2004.